

object to the application. Colwell does not teach or suggest that an interface such as the user interface of Colwell can return a command object such as an image viewer to an application such as an electronic mail application.

In view of the remarks noted above, the Applicant's representatives believe the rejections to base claims 1, 9, 16, and 23 have been traversed thereby placing claims 2-8, 10-15, 17-22, and 24-27 in condition for allowance in their present form for at least the reasons noted above.

In view of the above, Applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

If any fees are due in connection with the filing of this amendment, the Commissioner is authorized to charge such fees to Deposit Account 50-0388 (Order No. SUN1P123). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP



Godfrey K. Kwan
Registration No. 46,850

BEYER WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, CA 94704-0778
Tel: (650) 961-8300

6,021,433 (Payne). Claims 4, 11, and 18 were rejected under 35 U.S.C. §103(a) as being unpatentable over Colwell, in view of Payne, and in view of “The Java Language Enviroment.”

Colwell describes a system for searching and retrieving text stored in files. In response to a word search request initiated by a user, the system builds index files representing the approximate position and frequency of every word in the search request within files on a given storage unit (column 2, lines 18-20). A viewer module is invoked to view the file containing the words in the search request.

By contrast, Claims 1 and 23 explicitly recite “a data handler mechanism arranged to interface with an application.” It appears from the Examiner’s rejections that since the user interface module of Colwell interfaces with a user that this somehow discloses or suggests a data handler mechanism arranged to interface with an application, as required by claims 1 and 23. However, in no manner does Colwell teach or suggest a data handler mechanism arranged to interface with an application. The specification refers to an “application” in several places. According to specific embodiments, the application may be a file manager or an electronic mail application (page 15, lines 2-4). If the application is a file manager, then the names and the icons for files are collected and presented to the user (page 16, lines 11-14). It should be noted that in no manner should this clarification be construed to limit the scope of the claims to the embodiments described, as other embodiments fall within the scope of the claims. However, in this example, the application presents names and icons to the user. In no manner does Colwell disclose or suggest a data handler mechanism arranged to interface with an application such as an email application.

Claims 1 and 23 further recite that the data handler mechanism is in communication with the data retriever mechanism. Colwell neither discloses nor suggests a data handler mechanism arranged to interface with an application such as an email application and that is in communication with a data retriever mechanism.

Claims 9 and 16 explicitly recite “returning the command object to the application, wherein the interface returns the command object to the application.” As noted above, the application is distinct from a user. A command object, such as an image viewer, is returned to an application such as an electronic mail application, wherein the interface returns the command

APPENDIX

1. (Twice Amended) A computer-implemented framework for associating data with a command object, the command object being arranged to operate on the data, wherein the data is associated with an application, the computer-implemented framework comprising:

a data handler mechanism arranged to interface with the application;

a data retriever mechanism in communication with the data handler mechanism, the data retriever mechanism being arranged to obtain the data and to pass the data to the data handler mechanism; and

a mapping mechanism in communication with the data handler mechanism, the mapping mechanism being separate from the data handler mechanism, the mapping mechanism being arranged to obtain the command object, wherein the command object is obtained by the mapping mechanism based substantially on the data.

2. A computer-implemented framework according to claim 1 wherein the data is a stream of bytes, and the data handler mechanism is further arranged to bind the stream of bytes to the command object.

3. A computer-implemented framework according to claim 1 wherein the data retriever mechanism includes a data content handler mechanism in communication with the data handler mechanism, the data content handler mechanism being arranged to convert the data into a data object, wherein the data handler mechanism is further arranged to bind the data object to the command object.

4. A computer-implemented framework as recited in claim 3 wherein the data object is created using the Java™ programming language, and the command object is a Java™ command object.

5. A computer-implemented framework as recited in claim 1 wherein the data is one of text data and image data.

6. A computer-implemented framework as recited in claim 1 wherein the data handler is further arranged to receive a request from the application, to bind the data to the command object, and to return the command object to the application.

7. A computer-implemented framework as recited in claim 1 wherein the data retriever includes a data source mechanism arranged to obtain a stream of bytes and a data content handler mechanism arranged to convert the stream of bytes into a data object, the data source mechanism being in communication with the data content handler mechanism, wherein the data handler mechanism is further arranged to bind the data object to the command object.

8. A computer-implemented framework as recited in claim 1 wherein the mapping mechanism includes a look-up table arranged to associate the command object with the data.

9. (Twice Amended) A computer-implemented method for associating data with a command object in response to a request from an application, the method comprising:

- accessing the data through an interface in response to the request from the application, the interface being independent from the application and in communication with the application, wherein the request from the application is processed by the interface;

- accessing a mapping mechanism which is in communication with the interface, the mapping mechanism being independent from the application such that the mapping mechanism is not a component of the application, the mapping mechanism being maintained separately from the interface, the mapping mechanism further being arranged to locate a command object that is appropriate for the data, wherein the mapping mechanism is accessed by the interface;

- obtaining the command object that is appropriate for the data, wherein the mapping mechanism obtains the command object and passes the obtained command object to the interface;

- binding the command object to the data, wherein the interface binds the command object to the data; and

- returning the command object to the application, wherein the interface returns the command object to the application.

10. A computer-implemented method as recited in claim 9 wherein accessing the data through an interface includes:

- passing a stream of bytes to a data content handler mechanism arranged to create a data object from the stream of bytes; and

- passing the data object to the interface, wherein the data is the data object.

11. A computer-implemented method as recited in claim 10 wherein the data object is created using the Java™ programming language, and the command object is a Java™ command object.
12. A computer-implemented method as recited in claim 9 wherein accessing the data through the interface includes accessing a data retriever which is arranged to obtain the data, wherein the data is a stream of bytes.
13. A computer-implemented method as recited in claim 9 further including operating on the data using the command object.
14. A computer-implemented method as recited in claim 9 wherein the command object that is appropriate for the data is selected from a set of command objects associated with a command list, the command list being associated with the data, the method further including accessing the command list, wherein the command list is accessed by the interface.
15. A computer-implemented method as recited in claim 14 wherein accessing the command list includes receiving a request for a command list from the application, the request for the command list being received by the interface, wherein the interface performs the steps of:
 - obtaining a type associated with the data;
 - obtaining the command list through the mapping; and
 - returning the command list to the application.
16. (Twice Amended) A computer program product for associating data with a command object in response to a request from an application, the computer program product comprising:
 - computer code for accessing the data through an interface in response to the request from the application, the interface being independent from the application and in communication with the application, wherein the request from the application is processed by the interface;
 - computer code for accessing a mapping mechanism which is in communication with the interface, the mapping mechanism being independent from the application such that the mapping mechanism is not a part of the application, the mapping mechanism further being separately maintained from the interface, the mapping mechanism further being arranged to locate a command object that is appropriate for the data, wherein the mapping mechanism is accessed by the interface;

computer code for obtaining the command object that is appropriate for the data, wherein the mapping mechanism obtains the command object and passes the obtained command object to the interface;

computer code for binding the command object to the data, wherein the interface binds the command object to the data;

computer code for returning the command object to the application, wherein the interface returns the command object to the application; and

a computer-readable medium that stores the computer codes.

17. A computer-readable medium as recited in claim 16 wherein the computer program code devices configured to cause the computer to access the data through an interface include computer program code devices configured to cause a computer to execute the steps of:

passing a stream of bytes to a data content handler mechanism arranged to create a data object from the stream of bytes; and

passing the data object to the interface, wherein the data is the data object.

18. A computer-readable medium as recited in claim 17 wherein the data object is created using the Java™ programming language, and the command object is a Java™ command object.

19. A computer-readable medium as recited in claim 16 further including computer program code devices configured to cause the computer to operate on the data using the command object.

20. A computer-readable medium as recited in claim 16 wherein the command object that is appropriate for the data is selected from a set of command objects associated with a command list, the command list being associated with the data, the computer-readable medium further including computer code devices configured to cause the computer to access the command list through the interface.

21. A computer-implemented framework according to claim 1 wherein the command object is obtained by the mapping mechanism based substantially on the data without an external input from a user of the application.

22. A computer-implemented framework according to claim 1 wherein the command object is obtained by the mapping mechanism based substantially on the data without directly involving the application.

23. (Twice Amended) A computer-implemented framework for associating data with a command object, the command object being arranged to operate on the data, wherein the data is associated with a selected application, the computer-implemented framework comprising:

a data handler mechanism arranged to interface with a plurality of applications, the plurality of applications including the selected application, wherein the data handler mechanism is independent from the plurality of applications;

a data retriever mechanism in communication with the data handler mechanism, the data retriever mechanism being arranged to obtain the data and to pass the data to the data handler mechanism; and

a mapping mechanism in communication with the data handler mechanism, the mapping mechanism being substantially separate from the data handler mechanism, the mapping mechanism being arranged to obtain the command object, wherein the mapping mechanism is associated with the plurality of applications and is arranged to obtain the command object without directly involving the selected application.

24. A computer-implemented framework as recited in claim 23 wherein the mapping mechanism and the data handler mechanism are separately maintained.

25. (Amended) A computer-implemented framework as recited in claim 23 wherein the mapping mechanism is not a component of the data handler mechanism.

26. A computer-implemented framework as recited in claim 1 wherein the mapping mechanism and the data handler mechanism are separately maintained.

27. A computer-implemented framework as recited in claim 1 wherein the mapping mechanism is not specific to the application while the data handler mechanism is substantially specific to the application.